Implementation Experience with MANET Routing Protocols

Kwan-Wu Chin, John Judge, Aidan Williams and Roger Kermode
Sydney Networks and Communications Lab
Motorola Australia Research Centre
12 Lord St, Botany, NSW, Australia 2019
{*kwchin, johnj, aidan, rkermode*}*@arc.corp.mot.com*

ABSTRACT

This paper outlines our experience with the implementation and deployment of two MANET routing protocols on a five node, four hop, network. The work was prompted by the lack of published results concerning the issues associated with the implementation of MANET routing protocols on actual wireless networks, as opposed to results of simulation experiments. We examined implementations of two distance vector MANET routing protocols and found a number of problems with both protocols during the course of our experiments. The most significant was that neither protocol could provide a stable route over any multi-hop network connection. The route discovery process of both protocols is fooled by the transient availability of network links to nodes that were more than one hop away. Packets transmitted over a fading channel cause the routing protocol to conclude incorrectly that there is a new one hop neighbor that could provide a lower metric (hop count) route to even more distant nodes. This can occur even when nodes are stationary, mobility resulted in even less route stability. We implemented a simple signal strength based neighbor selection procedure to test our assertion that fading channels and unreliable network links were the cause of the failure of the routing protocols. The result was that neighbor discovery and the filtering for neighbors with which nodes could communicate reliably enables the creation of reliable multihop routes. Based on our experiences, we outline several recommendations for future work in MANET research.

1. INTRODUCTION

The term ubiquitous computing was coined by Mark Weiser to describe a state of computing in which users are no longer aware of computation being done [28]. The emergence of smart environments, where devices are embedded pervasively in the physical world, has sparked many new research areas and represents a step towards ubiquitous computing. To this end, researchers have begun to outline plans to achieve ubiquitous computing. For example, Basu et al. [3] advocate the vision of *power-up-n-play* for smart environments in which no predefined infrastructures are installed and, when powered up, the devices "intelligently" configure and connect themselves to other devices. Bhagwat et al. [4] also focus on the interoperability of sensor devices and present three research issues: (1) distributed algorithms for self-organizing devices, (2) packet forwarding, and (3) Internet connectivity.

Mobile **a**d-hoc **net**work (MANET) routing protocols play a fundamental role in a possible future of ubiquitous devices. Current MANET commercial applications have mainly been for military applications or emergency situations[25]. However, we believe that research into MANET routing protocols will lay the groundwork for future wireless sensor networks and wireless plug-n-play devices. The challenge is for MANET routing protocols to provide a communication platform that is solid, adaptive and dynamic in the face of widely fluctuating wireless channel characteristics and node mobility.

The paper discusses our experience while implementing and deploying two distance vector MANET routing protocols. We examined both a public domain implementation of the Ad Hoc On-Demand Distance Vector (AODV) [21] routing protocol and implemented our own version of the Destination-Sequenced Distance Vector (DSDV) [20] routing protocol. The choice of routing protocols was pragmatically based on what (little) was available at the time this work was carried out. The AODV implementation was the freely available MAD-HOC implementation [15]. This implementation was based on an earlier draft of the AODV protocol and includes some MAD-HOC specific extensions. Where AODV is referred to in this paper we mean the MAD-HOC implementation unless otherwise stated. At the time our work was carried out this was the only public domain MANET routing protocol implementation that had a license suitable for our use and that we could get to compile, run and work on our network. Faced with no other available public domain code and reluctant to base our work solely on one protocol implementation we coded an alternative. DSDV was chosen due to it's relative simplicity and the fact that it is a table based protocol rather than an "on demand" protocol like AODV. Our implementation was based largely on the paper by Perkins et al. [20].

Both protocols were deployed on a five hop, four node testbed based on Linux workstations and 802.11b wireless LAN cards configured to use the Lucent ad hoc mode. We found that neither protocol could provide stable multihop network routes. The main cause was the failure of the route discovery processes in provisioning for unreliable links which are inherent in wireless channels. The route discovery process was fooled by transient link availability with nodes that were too distant for reliable communication to take place. A couple of routing packets sent over this link is enough to temporarily fool the routing protocol into assuming a more direct (lower

hop count) route exists to the desired destination.

To test the assertion that transient link availability was the cause of the failure of the routing protocols we developed a signal quality based neighbor selection program called *powerwave*. The inclusion of *powerwave* for neighbor selection stabilized multi-hop routes for both routing protocols to the point where they could carry useful amounts of user data.

A number of extensive simulation studies on various MANET routing protocols have been performed by various researchers [25][5][16][8][7]. However, there is a severe lacking in implementation and operational experiences with existing MANET routing protocols. Previous implementation experiences include wireless Internet gateways (WINGS) [11], implementation of ODMRP [2], AODV implementation by Royer et al. [24] and ABR implementation by Toh et al. [27]. These studies only highlighted performance issues specific to the protocol being used. By far the most extensive implementation study to date was conducted by Maltz et al. [17] in describing their implementation of DSR.

Unlike previous work, our work reports on the experience of building an operational ad-hoc network that is capable of carrying useful data. We report several interesting observations not reported elsewhere for the use of MANET protocols within pico-cell environments. It is worthwhile noting that this paper's objective is to report on the operational feasibility of existing routing protocols and efforts undertaken to create a reliable ad-hoc network. In many ways this is a step back towards fundamental issues and away from the MANET routing protocol aspects usually examined in simulation studies. Whereas simulation studies commonly report on performance metrics such as throughput, latency and packet loss this paper reports on the fundamental issue of "do MANET routing protocols work". The answer is yes but, in the case of the two distance vector protocols we examined, only if the inherent unreliable and transient nature of wireless network links are taken into account.

This paper is organized as follows. In Section 2 we provide a brief summary of AODV and DSDV. This is followed by implementation details of both these protocols in Section 3. In Section 4 we describe the testbed used for our experiments. Section 5 presents the problems and observations gained from setting up the testbed and running the routing protocols over it. In Section 6, we present the workings of *powerwave*. Based on our experience with MANET routing protocols, we discuss issues and problems encountered in relation to existing routing protocols and propose some future directions in Section 7. Finally, the conclusions are presented in Section 8.

2. BACKGROUND

In this section we review the workings of the AODV and DSDV MANET routing protocols. Comprehensive reviews of other routing protocols are available in [25],[12] and [5].

AODV is characterised as an on-demand (also called reactive) routing protocol. Routes are created as needed at connection establishment and are maintained for the duration of the communication session. During route discovery a node broadcasts a route request (RREQ) message for a given destination address. Nodes that have a route to the destination respond to the RREQ by sending a route reply (RREP) message to the source and record the route back to the source. Nodes that do not have a route to the destination rebroadcast the RREQ message after recording the return path to the source. In the event of link breakage a route error (RERR) message is sent to the list of nodes (referred to as precursors) that rely on the broken link. Upon receipt of a RERR message, the corresponding route is invalidated and a new RREQ may be initiated by the source to reconstruct the route [21]. The time-to-live (TTL) field is used in RREQs for an expanding ring search to control flooding. Successive RREQs use larger TTLs to increase the search for destination node.

Unlike AODV, DSDV [20] is a table-driven (or proactive) routing protocol and is essentially based on the basic distributed Bellman-Ford routing algorithm [1]. Each node in the network maintains a routing table consisting of the next hop address, routing metric and sequence number for each destination address. To guarantee loop free operation, routing updates from a given node are tagged with a monotonically increasing sequence number to distinguish between stale and new route update messages. Nodes periodically broadcast their routing tables to neighbouring nodes. Given sufficient time, all nodes will converge on common routing tables that list reachability information to each destination in the network. Route updates are generated and broadcast throughout the network when nodes discover broken network links. Nodes that receive a route update check to see if the sequence number specified in the route update message is higher than the sequence number recorded in their own routing table before accepting the update. DSDV reduces routing messages overheads by supporting both full and incremental updates of routing tables.

The main characteristic of table-driven protocols is that a route to every node in the network is always available regardless of whether or not it is needed. This results in substantial signaling overhead and power consumption [25]. Furthermore, table driven protocols transmit route updates regardless of network load, size of routing table, bandwidth and number of nodes in the network [5]. Interested readers are referred to Toh et al. [25] for a qualitative comparison based on simulation experiments between flavors of both on-demand and table-driven routing protocols.

3. ROUTING PROTOCOL IMPLEMENTATIONS

This section presents implementation details of the AODV and DSDV protocols used in our experiments and provides a background to the discussions and observations which will follow regarding the deployment and implementation issues we have encountered.

3.1 MAD-HOC Implementation of AODV

The AODV routing protocol used in our experiments was implemented by the MAD-HOC group [15] and can be obtained from *http://mad-hoc.flyinglinux.net* [1]. There are two

[1]At the time of our experiments there were two publicly available MANET routing protocols, CMU's DSR and MAD-HOC's AODV. We chose MAD-HOC's AODV over

main components to the MAD-HOC implementation: (1) *packet_capture* and (2) *aodv_daemon*.

The *packet_capture* program captures packets that traverse the network interface and triggers the *aodv_daemon* when particular packets are seen. The capture mechanism is implemented using the libpcap library [14]. Three types of packets are of interest: address resolution protocol (ARP) packets, Internet control message protocol (ICMP) packets and Internet protocol (IP) packets. Un-answered ARP requests from a host indicate that a route to a given destination is required, *packet_capture* extracts the destination IP address from the ARP packet, and passes the address to the *aodv_daemon*. *aodv_daemon* then generates a route request for the destination. When an ICMP message is parsed *packet_capture* determines whether the ICMP message received is of type ICMP DEST UNREACH, ICMP UNREACH HOST or ICMP UNREACH HOST UNKNOWN. If the message matches the above ICMP types, the *aodv_daemon* is notified of a link breakage to a given destination address. All other ICMP messages are ignored. When a link break is detected, the *aodv_daemon* issues a route error message to all hosts using the broken link. The source address of data packets intercepted by *packet_capture* are passed directly to *aodv_daemon* to update the route lifetime which the data packets arrived on. The MAD-HOC AODV implementation used hello messages, periodic broadcasts, to maintain a local connectivity list.

The main problem with the MAD-HOC AODV implementation was that buffering was not performed while route construction was in progress. In practical terms, we found that a *telnet* session had to be initiated multiple times before a session could be established. When running ping over a four hop route, with the default one second gap between successive pings, the first five packets were usually lost before the route was successfully established.

3.2 DSDV Implementation

The second routing protocol we chose to experiment with was DSDV. The choice was made due to DSDV's simplicity, thus enabling us to easily code up and debug the operation of DSDV on our testbed. DSDV's simplicity proved valuable during our experimentation especially when explaining the poor operation of DSDV on our testbed.

Our DSDV implementation was based on the ACM SIGCOMM'94 paper by Perkins et al. [20] with the addition of a neighbor handshake protocol to check for bi-directional links. Our DSDV implementation used the Multi-threaded Routing Toolkit (MRT) [19] for platform independence and for interfacing with the kernel routing table, socket and file input/output (IO). In addition, MRT also provided some convenient data structures for holding information regarding machine interfaces and utilities for manipulating IP addresses. Due to the small scale of our testbed, the incremental update aspects of DSDV were not implemented (all the routes could easily fit in the one packet). The hysteresis timers were also not implemented as we did not have many alternate routes of the same hop count.

CMU's DSR due to extensive documentation, and hardware and operating system compatibility with our testbed.

3.2.1 The SEEN Metric and State

The original paper describing DSDV [20] specified that DSDV assumes bi-directional links but does not include any mechanism for ensuring a link was bi-directional before a route was put in place. It was found that such a mechanism was crucial with fading channels. We extended DSDV through the inclusion of a handshake protocol that makes use of the SEEN metric to signal that a new neighbor had been detected.

The SEEN metric was defined as an integer value outside the range of one to INFINITY[2]. DSDV nodes advertise a route to a node with $metric = SEEN$ on the reception of a packet from a neighbor for the first time. All other nodes, apart from the node listed as the route destination, ignore this route. On receiving a routing advertisement for itself with a $metric = SEEN$ a node makes and advertises a route to the sending node. Nodes will only advertise a route to another node with a SEEN metric for a short period of time, if no reciprocal route advertisement is received then the SEEN state times out and the route is no longer advertised. The signaling process used in the discovery of a bi-directional neighbor using the SEEN metric is illustrated in Figure 1.

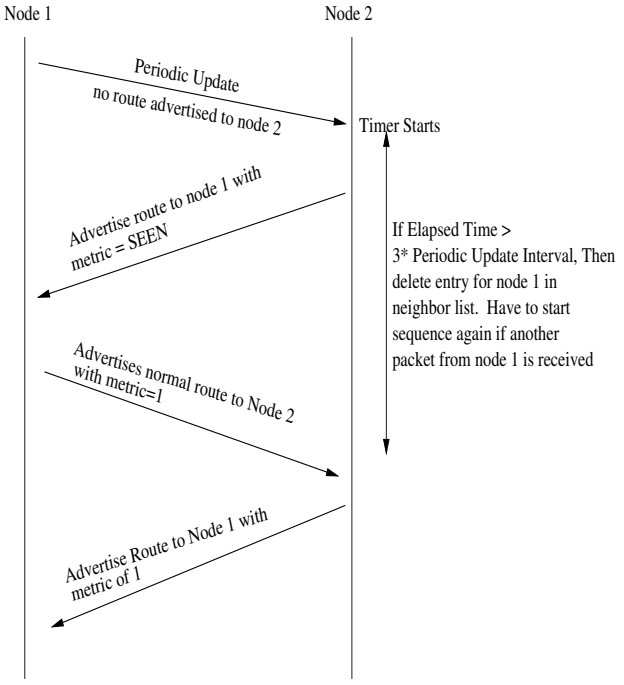

Figure 1: DSDV's Signaling Process Using SEEN Metric, No Existing Route Between Node 1 and Node 2

4. TESTBED

Figure 4 shows the network topology of our testbed. Our testbed consisted of two notebooks and three desktop computers, equipped with Lucent Wavelan IEEE 802.11b PCMCIA cards and running Linux (Debian with 2.2.15 kernel). We used version 6 of the Linux driver from Lucent for the IEEE 802.11b cards, with the transmit rate set to 1 Mb/s

[2]INFINITY itself was defined as 16 and is used to signal that a destination is no longer reachable

and the operation mode set to ad-hoc[3]. The lowest channel rate was chosen to avoid the cards stepping down transmission rates automatically (a feature that we could not otherwise disable). The cards were configured to transmit on an otherwise unused channel to avoid interference from other IEEE 802.11b devices in our lab. To limit the transmission range, we wrapped each card with a metallic anti-static bag. As a result, we managed to drop the transmission range from 250 meters to approximately five meters. This enabled us to create a four hop network in our lab and avoid the problem of having to locate the experiment in a large field.

It is important to note that the anti-static wrapping did not alter the radio propagation characteristics of an indoor office environment consisting of soft partitions. The observed radio propagation behavior, i.e., Rayleigh Fading, of the testbed is consistent with Hashemi [13]'s study on indoor radio propagation models. Figure 2 and 3 show a comparison of the signal-to-noise ratio as measured on our testbed and that of Rayleigh fading respectively. As can be seen, both experimental and theoritical model agrees, hence the anti-static wrapping did not alter the fading behavior of the channel which contributes to transient links. Readers who are interested in indoor radio propagation models and Rayleigh fading are referred to [13] and [23].

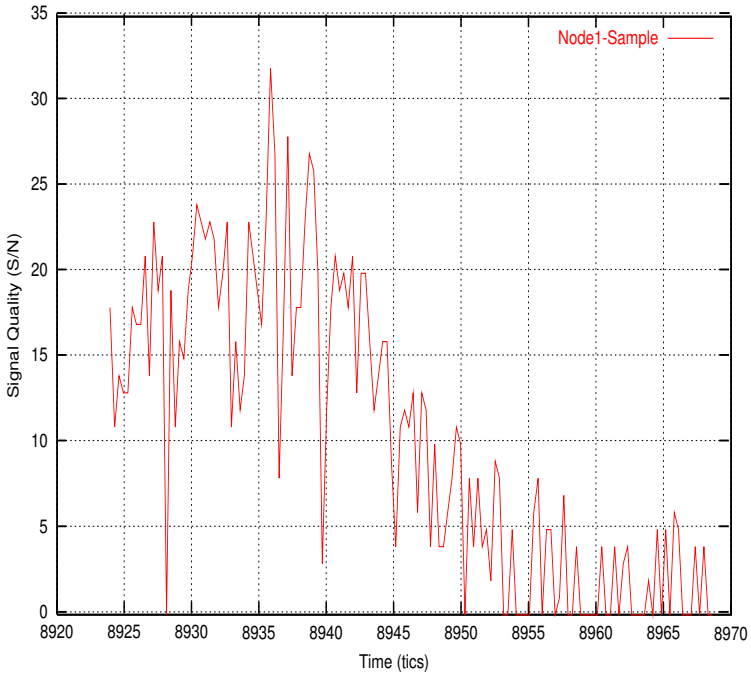

Figure 2: Signal Level Measurements from Our Testbed, Soft Partition Office.

In order to verify we have a working ad-hoc network we ran the following experiments. The first experiment consisted of an application residing on mobile host 2 (MH_2) that transmits UDP packets to the *discard* service on MH_1. We then monitored the number of packets transmitted and received as MH_2 moved along the line of hosts toward, or away from $MH1$. Motion towards MH_1 was referred to as "downstream" while motion away from MH_1 was referred

[3]Lucent's propriety ad-hoc mode

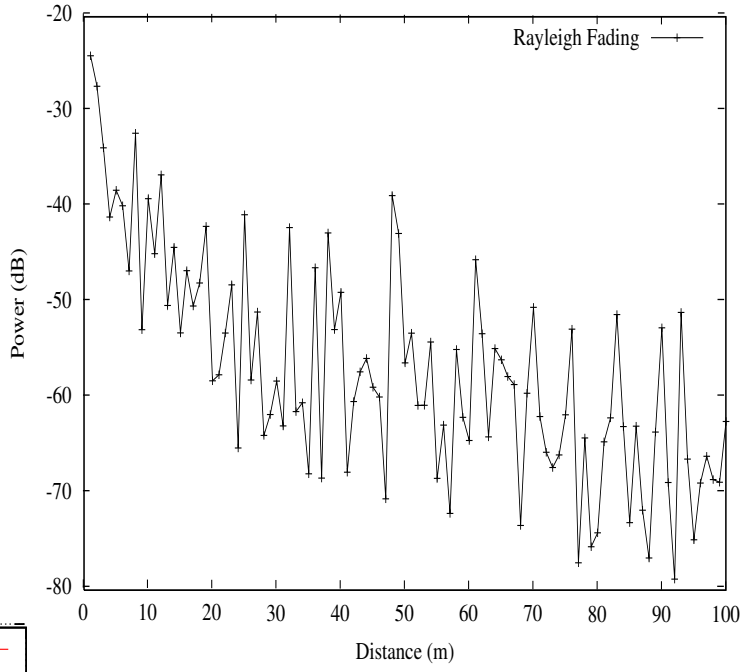

Figure 3: Rayleigh Fading. The figure was generated by calculating the received power when two nodes starting at distance 0m, and then calculating their received power after moving them apart at increment of 5m.

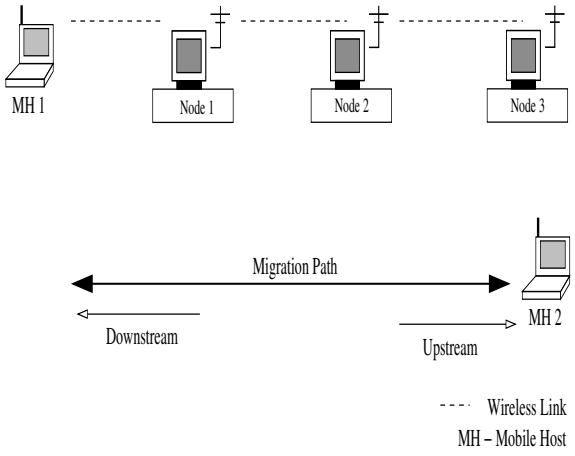

Figure 4: Testbed Topology

to as upstream. In the second experiment, we performed file transfers (using FTP) between MH_1 and MH_2. In our experiments no other sessions were present and the network traffic in our experiments consisted entirely of data transfer between the mobile nodes and routing messages. Moving MH_2 along the line of nodes exercised the adaptive features of the routing protocols. The nodes were placed such that MH_2 should route packets through each of $node1$, $node2$ and $node3$ in turn as it is moved upstream. Each of the fixed nodes was placed so that it could communicate reliably with adjacent neighbors but could not send or receive packets reliably to the other more distant fixed nodes.

5. EXPERIMENTAL OBSERVATIONS

5.1 Fading and Transient Network Links

It was found that transient radio links resulted in poor operation of both the routing protocols examined where no reliable routes could be established. The poor operation was due to the creation and maintenance of routes without taking the stability, or quality, of the network links comprising the route into account. The fundamental problem was that successful transmission of a datagram over a wireless network link is probabilistic, regardless of lower level protocols. In practice this probabilistic effect became evident in two ways; occasional dropped packets on a normally "good quality" network link and occasional successful packet transmissions on a normally "poor quality" network link. We found that the occasional dropped packet did not present much of a problem for either of the routing protocols examined. On a "good" network link the link layer acknowledgements in 802.11 replaced lost unicast packets and the routing protocols appeared to be able to handle the occasional lost broadcast, or multicast, packet. In contrast the occasional appearance of a channel between two nodes that could not normally communicate was disruptive to the routing protocols on our testbed. The problem manifested itself in the creation of network routes that were not suitable for the reliable transmission (and reception) of user data. These routes were chosen over other route options by the protocols selecting for lowest hop routes, regardless of any sort of measure of route quality. As stated in the introduction a similar effect for the DSR routing protocol has been observed on another testbed [18].

We found that it was practically impossible to establish a stable *telnet* session between nodes over a three or four hop route on our testbed. For example when using the topology described in Figure 4, we found that $Node_1$ could still detect $Node_3$'s signal occasionally despite careful placement and orientation. As a result we observed that both nodes would randomly receive a packet from the other. If AODV was engaged in a route building process it would use the unreliable one hop route from $Node_1$ to $Node_3$ in preference to the two hop alternative. DSDV would replace the existing two hop route between the nodes with the unreliable one hop route. Very little user data would be transmitted over this unreliable route and user sessions would hang pending the reestablishment of the more reliable two hop route.

In a related work, Maltz et al. [17] reported similar behavior while building a MANET testbed and experimenting with Dynamic Source Routing (DSR) routing protocol. The following modifications to DSR were suggested to overcome the problem of routing over unreliable links: (1) monitor route error on links, (2) use the geographic positioning system (GPS) to determine the neighbor proximity (assuming physical proximity will provide the best channel) and (3) combine GPS with route error monitoring. Reliability was tested over a three node, two hop network with the nodes arranged in a line. The network included packet filtering software to prevent packets from being transmitted directly from one end node to the other. They found that an FTP file transfer between the end nodes was more reliable when the packet filtering software was enabled. Ramanathan et al. [22] also reported problems with transmission range when testing out their quality of service (QoS) based routing protocols. However, no solutions to unreliable links were suggested.

Published articles reporting on MANET routing protocol performance often rely on simulation experiments. Experiments run on our testbed uncovered considerable difference in the probability of successfully receiving packets on a MANET node versus the probability of successful packet reception in some simulation environments. In a simulation environment, such as *ns-2* [10], it is generally assumed that the probability of receiving a packet is effectively one (pending collisions etc) and once a node moves out of another node's signal range, or a given distance, this drops to zero. However, our experiments have shown that this is unrealistic; signals tend to decay slowly and there is no cutoff point. We suspect that the use of simplistic radio propagation models in MANET simulation environments has led to inaccurate assessments of the performance of various routing protocols, especially those which utilize hop count as the dominant route selection metric. Thus, one area for future work is the incorporation of better radio propagation models that support channel fading and other inputs to the probabilistic nature of wireless channels. For example, Rappaport [23] lists a number of factors that affect fading in an in-door environment such as multi-path propagation, mobile node speed, surround object speed and signal bandwidth.

5.2 Handoff in a MANET

In conventional cellular networks, the signal-to-noise ratio (SNR) of the connection between mobile phone and base stations is monitored to determine when to hand off from one base station to another. In a MANET, current protocols do not predict when a link's SNR will fall below a threshold. The periodic HELLO messages in AODV and route update timers in DSDV are not used to anticipate hand off, they indicate presence or absence of a neighbor node. Consequently, the route maintenance process at both AODV and DSDV is only initiated after link breakage already ocurred.

DSDV behaves differently depending on the mobile nodes direction of movement. DSDV pro-actively changed to a lower hop count route if one was available, but hung on to a route until it is explicitly broken should a lower hop count route not be available. The effect with DSDV was smooth handover when MH_2 (in Figure 4) was moving downstream but no handover in the upstream direction.

In the upstream direction two things would prompt a new (higher hop count) route to be used. First, the connection to the previous fixed node would have to timeout prompting

a switch to the next best available route being advertised by the new neighbor. Or second, the link between the previous fixed node would have to break along with a route advertisement being received from the new neighbor with a higher hop count and a higher sequence number. The new sequence number would then invalidate the old route and cause the new route to be used instead.

5.3 AODV Specific Issues

5.3.1 Pico cell size and AODV's timers

A problem encountered were AODV's default parameters. Since the transmission range of each node was reduced in our testbed to less than 5m, we had in effect constructed a network with pico sized cells. In this environment the default MAD_HOC AODV timers unnecessarily prolonged route construction and required tunning before an acceptable performance could be achieved. The parameters we changed are listed on Table 1. AODV's parameters as specified in [21] are left to the implementors, however recent drafts have used more conservative parameters than those in the MAD-HOC implementation shown in Table 1.

BCAST_ID_SAVE is used to prevent over flooding of RREQ messages. When a new RREQ is intercepted, the information within the RREQ is recorded and the information is added to an interval queue along with a time interval (current time plus BCAST_ID_SAVE). In the event of another RREQ appearing within this time interval, the RREQ is discarded.

RREQ_RETRIES bounds the number of RREQs for a given destination. The default value is two. We found this value to be too conservative, and found that five was more appropriate value.

ACTIVE_ROUTE_TIMEOUT is used to determine the lifetime of a given route. The lifetime of each route maintained by a given node is refreshed after observing data packets or HELLO messages on that route. In a pico-cell environment, the default value needs to be small. In our testbed where nodes moved at slow walking pace, the time for a node to traverse given cell was around five and we found a route timeout value of one second was appropriate.

Both NODE_TRAVERSAL_TIME and NET_DIAMETER had to be modified to suit our network topology. The NODE TRAVERSAL TIME was modified to increase the route construction time. The default value of NET_DIAMETER was set to 35 nodes and this was changed to five to reflect the number of nodes in our testbed.

The last parameter to be modified was ALLOWED HELLO LOSS which determines how many HELLO messages are lost before a link is considered broken. Routes were timing out frequently in our testbed and we set the ALLOWED HELLO LOSS parameter to five to increase stability.

The optimization of AODV by changing the parameters to suit our testbed was done on a trial and error basis. To date there are no published guidelines or heuristics for setting AODV's parameters or adapting them to a given network. The parameters shown in Table 1, and the other AODV parameters that have been defined in the AODV specification [21], would most likely have to be modified for use in other networks.

5.3.2 ARP Interactions

The reliance of the MAD-HOC AODV implementation on sniffing ARP packets to signal the need for route construction led to two problems. The first problem was that packets were not buffered while the route was being built. As mentioned in Section 3 this led to packets being dropped and the need to start an application such as *telnet* a number of times before a route was actually built. The second problem was that a route will never be constructed if there is an entry in the ARP cache. Spurious ARP cache entries exist for one or more reasons. Either the two nodes in question had once been adjacent, and the ARP cache entry had yet to time out, or an ARP reply was un-expectedly received from a remote node (over an unreliable link) and the cache then prevented a more reliable route being found.

One work around to these problems was to regularly flush the ARP cache and to start applications multiple times while waiting for the route building process to complete. In practice this would be achievable by using ping and waiting for a successful reply before starting the intended application. A better solution is the one proposed in [24] that uses a *netlink* socket to communicate routing information with the kernel space and a dummy route for buffering data packets pending route construction.

5.4 DSDV

5.4.1 Route Stability

The first thing we noticed about our DSDV implementation was its relative stability compared to the MAD-HOC's AODV implementation. DSDV was less affected by unreliable connections to distant nodes. This was mainly due to the use of the SEEN metric (requiring a handshake before the link would be used in routes) and less interaction with the ARP cache as the routing table was pre-populated with host routes (negating the need to ARP).

However DSDV was adversely affected by transient link availability. Even when all the network nodes were stationary the routing table would slowly "churn" as routes were constructed to distant nodes and then timeout.

6. SIGNAL QUALITY BASED NEIGHBOR SELECTION

Our observations/experiments showed that the main shortcoming with both AODV and DSDV to be a failure to handle the unexpected availability of a channel to a distant node. The subsequent use of one hop links to distant neighbors resulted in unreliable routes over which very little user level data could be sent. The cause of this problem was the failure of the routing policy deamons in each node to differientiate between "good" and "bad" one hop neighbors. We hypothesized that if nodes could filter for reliable one hop neighbors and use only these neighbors as next hop gateways, the resultant routes should be reliable.

To verify our hypothesis we implemented a neighbor selection based on signal strength (called *powerwave*). We found that its use resulted in reliable multi-hop connections on our

Parameters	Default Values	New Values
BCAST_ID_SAVE	30000ms	3000ms
RREQ_RETRIES	2	5
RREP_WAIT_TIME	$(3 \times NODE\_TRAVERSAL\_TIME \times NET\_DIAMETER)/2$	No Change
NODE_TRAVERSAL_TIME	100ms	10ms
NET_DIAMETER	35	5
ACTIVE_ROUTE_TIMEOUT	9000ms	1000ms
ALLOWED_HELLO_LOSS	2	5

Table 1: *MAD-HOC's AODV Parameters*

testbed, and proves that neighbor selection is desirable and probably necessary in MANET environments.

6.1 Signal Based Route Selection

The *powerwave* implementation of neighbor selection was developed to be transparent to the routing protocol and used packet filtering to block routing messages from neighbors deemed unreliable. With neighbor selection we wanted to identify nodes one hop distant to which packets could be reliably sent and and make these available to the routing daemon.

Operating as a sublayer beneath the routing protocols assisted routing protocols in selecting routes over reliable network links. Our aim was to provide a generic neighbor discovery framework that we could use to test implementations of MANET routing protocols.

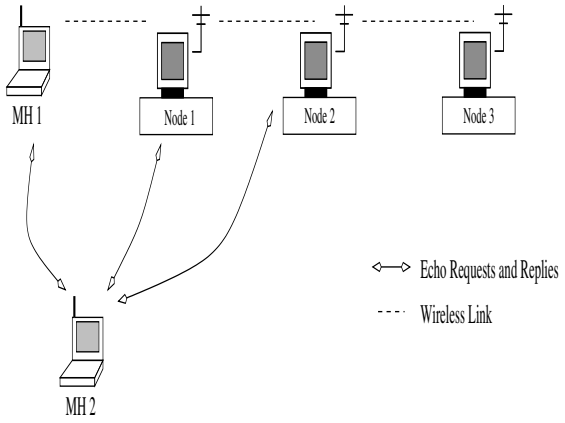

Figure 5: Measuring Signal Quality

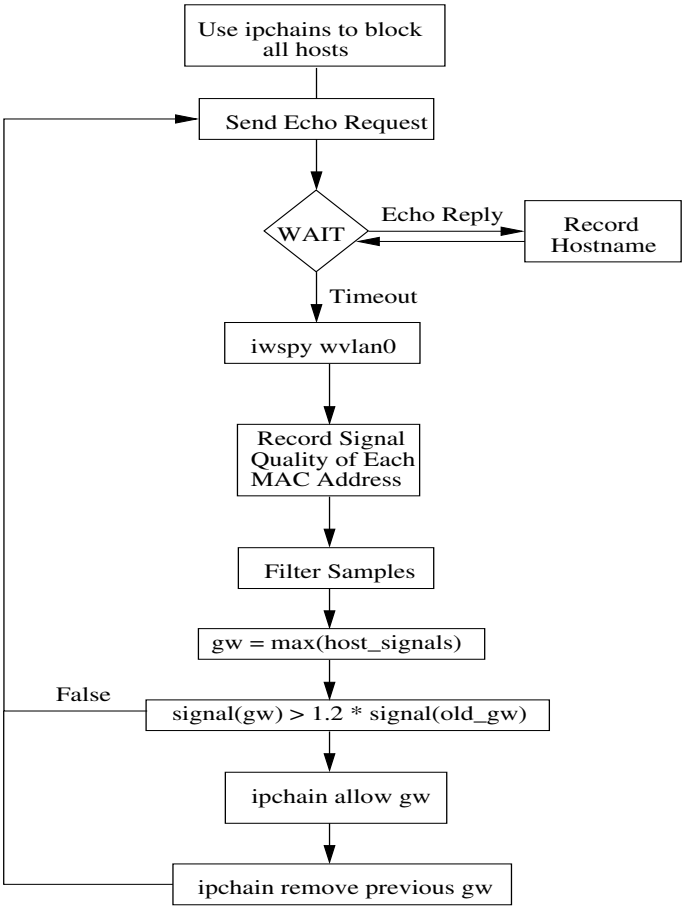

Figure 6: Flow-chart for Powerwave (Mobile)

Figure 6 shows the workings of our *powerwave* program on the mobile node. The value 1.2 was derived from measuring the signal strength on our testbed and determining an appropriate threshold that constitutes good signal strength. Before the program starts, the following *ipchains* rule is executed to filter out all messages (for AODV):

```
ipchains -A input -p udp -d 255.255.255.255 1303 -s 0.0.0.0 -j DENY
```

After the ipchains rule has been executed, echo requests were broadcasted and the SNR of replies were gathered. The signal strength associated with each link-layer address was then recorded and averaged. Averaging was required due to the random nature of a single SNR sample. Figure 7 shows raw SNR samples versus a moving average. The 'best' gateway[4] to route packets through was calculated based on previously recorded signal quality compared to current signal quality for each responding node. Note that the signal qualities used for comparison were averaged values. We tried using a fixed threshold value (20 dB) to determine the change of gateway. However, we found that due to the varying signal quality from multiple nodes, the choice of gateway tended to fluctuate frequently. Simply using a threshold value on the received signal quality was not effective and we found it did not yield reliable routes. Once the best gateway to route packets through was found, the following *ipchains* rule was executed (for AODV) to allow HELLO messages from the gateway:

```
ipchains -R input 1 -p udp -d 0/0 1303 -s ! %s -j DENY
```

We found that the *powerwave* program was also required at

[4]Next hop node through which to route outgoing packets

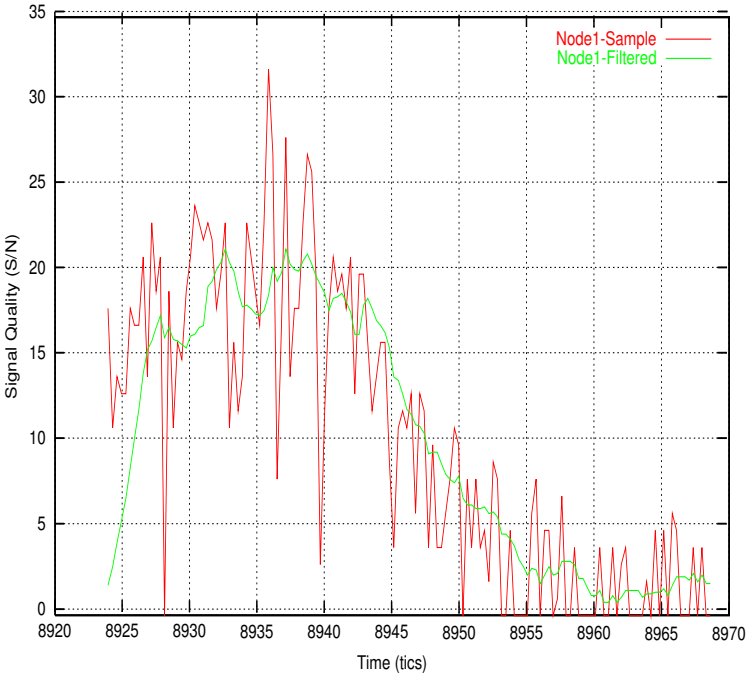

Figure 7: Sample vs. Filtered Signal Quality

stationary nodes in our testbed[5]. To ensure reliable links to their neighbors and more importantly to filter out HELLO messages from MH_2 that were transmitted over unreliable links. The reasons why *powerwave* was required on the static nodes were as follows. During route construction, a node downstream may have a shorter hop count, due to HELLO messages from MH_2, hence a RREP would be returned directly to MH_2 instead of being routed through the designated gateway. Since MH_2 ignores RREP messages from all nodes except for the designated gateway, MH_2 would then conclude that a route to MH_1 was impossible, resulting in the cancellation of the route construction process.

Powerwave programs running on stationary nodes required the following modifications:

- *Ipchain rules.* In the static nodes, specific rules were used to block out HELLO messages from non-neighboring nodes. For example, $Node_2$ (from Figure 4) only needs to listen to $Node_1$ and $Node_3$. The corresponding *ipchain* rules used to block out the appropriate nodes on $Node_2$ (for AODV) were:

```
# clean everything out
ipchains -F
# default deny
ipchains -A input -p udp -d 0/0 1303 -j DENY
ipchains -I input 1 -p udp -s node1 --dport 1303 -j ACCEPT
ipchains -I input 1 -p udp -s node3 --dport 1303 -j ACCEPT
# set up rule to be replaced blocking AODV from mobile
ipchains -I input 1 -p udp -s 10.1.0.100 --dport 1303 -j DENY
ipchains -L
```

  The *ipchains* configurations shown above are static which is unrealistic in a MANET where all nodes may move. However, the above rules can be adapted easily to moving nodes by imposing them dynamically using sampled SNRs of packets from neighboring nodes.

- *No echo request broadcast.* Echo requests were not needed since each node can read the signal quality of the echo request emitted by MH_2.

- *Interested in MH_2 only.* In our experiments, stationary nodes were only interested in receiving packets from MH_2. Once MH_2 is in range (quality above a given threshold) an *ipchains* rule was executed to allow routing packets to be passed to the routing daemon.

- *Thresholding.* The thresholding mechanism at stationary nodes was different to how thresholding was done at MH_2, where a fixed value was used instead of using a percentage of the averaged signal quality over time. To determine the threshold value at $Node_1$ to $Node_3$ and MH_1, graphs of SNRs collected from *powerwave* program were plotted. From these graphs, we determined a suitable threshold value, 10 dB. Thus if the signal quality of MH_2 exceeded 10 dB, *ipchains* was executed to allow the receipt of packets from MH_2. This threshold value was an arbitrarily selected value that was dependent on our network configuration. Determining an adaptive method that does not use thresholding is the subject of future work.

The *powerwave* program suffers from two shortcomings: (1) inefficient bandwidth consumption, and (2) inefficient interaction with AODV and DSDV. In the first case, *powerwave* on MH_2 broadcasts a continuous stream of echo messages in order for it (and other nodes) to measure the signal strength of packets received from each node. This increases contention time of other nodes wishing to transmit thereby reducing throughput of the network. In the second case, *powerwave* relies on blocking of HELLO messages from "bad" neighbors. Merely blocking routing messages leaves detection of broken links to the protocol timers. In future revisions, *powerwave* will signal the loss of a neighbor and also the appearance of a new neighbor directly to the routing protocol. Thereby routing protocols can be made aware of link-breakages and new neighbors in a timely manner.

While AODV and DSDV choose routes based on hop count, there are some MANET routing protocols such as SSA [9] that choose routes based on signal quality. Our experience with *powerwave* showed that a signal quality based routing protocol has to incorporate some form of stability metric after a route has been established to avoid the transfer of route as soon as a better signal link becomes available.

A similar approach to *powerwave* was also taken by Maltz et al. [18] where a program called *macfilter* was developed to filter out traffic from unwanted MAC addresses. A novel usage of *macfilter* was the emulation of a MANET where multiple nodes could be placed closely together and the signals from neighboring nodes filtered appropriately to give a different topology. The main difference between *macfilter* and *powerwave* is that *powerwave* uses SNR to dynamically determine which IP addresses to filter out whereas *macfilter* is statically configured for the topology in question.

An interesting conclusion from Matlz et al.'s work was that

[5]See Figure 4

they found neighbor selection to be important [18]. Our work further reinforces this believe, and we envisage more research work in the development of neighbor selection in MANET research.

7. DISCUSSIONS AND FUTURE WORK

7.1 Unstable Links

The majority of MANET routing protocols described in the literature were designed to handle topology changes and do not take unreliable links into account. Currently, only signal stability based adaptive routing (SSA) [9], ABR [26], and longest life routing protocol (LLRP)[29] support the notion of reliable routes. The route metrics use by SSA are average signal strength and route stability. By using these route metrics, packets will always be routed through the most reliable route (possibly closest node). Thereby route reconstruction cost is reduced and reliability of established route increases [9].

Unlike SSA, ABR only use route stability as the routing metric. Route stability is defined as the number of HELLO messages observe from a given neighbor. Hence, a neighbor with a given HELLO message count is considered stable. In both SSA and ABR, the destination has to choose the best route to take from a number of alternatives recorded from the various route requests received [29]. Further, once a route is setup there are no considerations for degraded links along the route. Routes are only rebuilt once they are broken.

The immediate future work is to re-evaluate existing hop based routing protocols with the addition of unreliable links.

7.2 Smooth Handoff

The notion of smooth handoff in MANET routing protocols has generally been overlooked. Improvements may be made by intelligently monitoring surrounding neighbors and determining whether a given node is able to prime an upstream/downstream node with a route to the destination. We found that a relatively smooth handover could be achieved by generating regular RREQs from MH_2. In other words, when a node detects a new neighbor a special message could be sent to prime the new neighbor, with routes to other new receiver nodes without waiting for existing routes to break.

Pro-active route construction will cause unnecessary traffic and duplicate routes which may then lead to the difficulty of removing invalidated routes. Further, the problem becomes more complicated if mobility is taken into account. Unlike traditional one hop wireless networks (e.g., cellular) where base-stations are fixed, the handoff decisions in MANETs are much more complicated.

It is interesting to note that the *powerwave* neighbor selection process had the side-effect of enabling a degree of handoff. The neighbor selection process filtered out neighbors before the network link disappeared entirely. User datagrams could still be forwarded over the link while the routing policy engine was finding a new route. It worked in our implementations because the routing parameters and the rate at which MH_2 moved matched.

7.3 Topology Dependent Parameters

Our experiments showed that the protocol parameters in both MAD-HOC's AODV and DSDV required some tuning before they would work properly. The determination of suitable timer values depended on channel rates, network topologies and mobility patterns [8]. The impact of these parameters on the performance of upper layer protocols is left for future work.

One method to allow for adaptive parameters is to introduce additional information. Protocols may rely on GPS, for example location aided routing protocols, to gather more information such as network topology and nodes proximity. Once the range of adjacent nodes are estimated, parameters may be adjusted accordingly.

7.4 Neighbor Selection Sub-Layer

The Internet MANET encapsulation protocol (IMEP) [6] is a mechanism to aggregate and encapsulate control messages. Also, IMEP provides a generic multi-purpose layer containing various common functionalities for MANET routing protocols. However, in the IMEP specification no consideration for signal strength was presented. It may be possible to use IMEP for filtering neighbors based on link stability rather than just to list neighbors that are in range.

Given the observations obtained from our experiments, one possible area of work is to extend upon IMEP's functionalities to incorporate mechanisms to shield wireless defects, and also offer various routing metrics which could be used by routing protocols.

8. CONCLUSION

In this paper we have outlined our implementation and deployment experiences with MAD-HOC's AODV and DSDV. Our experiments have provided insights into the real world deployment of MANETs and highlight issues that require further investigation. These are:

1. *Handling unreliable/Unstable links.*
2. *Minimizing the dependacy on topology specific parameters.*
3. *Mechanisms for handoff and reducing packet loss during handoff.*
4. *Incorporating neighbor discovery and filtering into a neighbor selection sub-layer.*

The first issue is a result of the current prevailing MANET protocol development/testing environments which appear to consist almost entirely of simulation experiments using *ns-2* and Glomosim. In implementing two MANET routing protocols, rather than simulating them, we discovered that the variability of networking conditions in the radio environment was such that the routing protocols did not work as reported in the literature. This led to the development of *powerwave*, and it was found that neighbor selection is crucial in the operation of MANET routing protocols. We believe our observations pertaining to unreliable/unstable links are not restricted to MAD-HOC's AODV implementation given that current AODV specification relies on hop

count and does not take into account the reliability of a given route or link.

The second issue is specific to a given routing protocol. As argued, having pre-configured parameters for a given topology is inappropriate given the inherent dynamic nature of MANETs, and affects the operation of routing protocols. Therefore, methods for adaptive adjustment of these parameters are required.

On the third issue, current MANET routing protocols do not appear to consider pre-emptive route construction based on signal strength in a similar way to how handoffs are done in cellular networks. We have observed that knowing whether a node is going upstream or downstream has added benefit. The concept of handoff, from one route that has a high probability of near term breakage to another route which is more stable is a possible area for future research.

Finally, there is scope for the development of a neighbor selection sub-layer like IMEP that incorporates a range of metrics that could be used by routing protocols. Various filters and heuristics could be developed which will be beneficial to MANET routing protocols.

9. ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for the constructive feedbacks on the presentation and content of this paper.

10. REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows, theory, Algorithms, and Applications*. Prentice-Hall, 1993.

[2] S. H. Bae, S.-J. Lee, and M. Gerla. Unicast performance analysis of the ODMRP in a mobile ad-hoc network testbed. In *Proceedings of IEEE ICCCN'2000*, Las Vegas, USA, 2000.

[3] P. Basu and T. D. C. Little. Task-based self-organisation in large smart spaces:issues and challenges. In *DARPA/NIST/NSF Workshop on Research: Issues in Smart Computing Environment*, Atlanta, USA, 1999.

[4] P. Bhagvat, C. Bisdjikian, P. Kermani, and M. Naghshineh. Smart connectivity for smart spaces. In *DARPA/NIST/NSF Workshop on Research: Issues in Smart Computing Environment*, Atlanta, USA, 1999.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad-hoc network routing protocols. In *Proceedings of the 4th ACM/IEEE Internation Conference on Mobile Computing and Networking (MOBICOM'98)*, Dallas, Texas, Oct. 1998.

[6] M. S. Corson and V. Park. An internet MANET encapsulation protocol (IMEP) specification. Internet Draft: draft-ietf-manet-imep-spec-00.txt, Nov. 1997.

[7] S. R. Das, R. Castaneda, and J. Yan. Simulation based performance evaluation of mobile, ad hoc network routing protocols. In *Proceedings of Seventh International Conference on Computer Communications and Networks (ICCCN'98)*, 1998.

[8] S. R. Das, C. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad-hoc networks. In *Proceedings of IEEE INFOCOM'200*, Tel-Aviv, Israel, 2000.

[9] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi. Signal stability based adaptive routing (SSA) for ad-hoc mobile networks. *IEEE Personal Communications*, 4(2):36–45, Feb. 1997.

[10] K. Fall and K. Varadhan. The VINT project. *ns* notes and documentation. http://www.isi.edu/nsnam/ns/.

[11] J. J. Garcia-Luna-Aceves, D. Beyer, and T. Frivold. Wireless internet gateways (WINGS). In *Proceedings IEEE Milcom'97*, Monterey, CA, 1997.

[12] M. Gerla, G. Pei, and S. J. Lee. Wireless, mobile ad-hoc routing. In *IEEE/ACM FOCUS*, New Brunswick, USA, May 1999.

[13] H. Hashemi. The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7), July 1993.

[14] Lawrence Berkeley National Lab. Libpcap: User-level packet capture library. ftp://ftp.ee.lbl.gov/libpcap-0.4.tar.Z, Feb. 1997.

[15] F. Lilieblad, O. Mattsson, P. Nylund, D. Ouchterlony, and A. Roxenhag. MAD-HOC AODV Implementation. Telecommunications Systems Lab, Techical Report. http://fl.ssvl.kth.se/.

[16] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson. The effects of on-demand behavior in routing protocols for multi-hop wireless ad-hoc networks. *IEEE Journal on Selected Areas in Communications special issue on mobile and wireless networks*, Aug. 1999.

[17] D. A. Maltz, J. Broch, and D. B. Johnson. Experiences designing and building a multi-hop wireless ad-hoc network testbed. Techical Report, CMU-CS-99-11, Mar. 1999.

[18] D. A. Maltz, J. Broch, and D. B. Johnson. Lessons from a full-scale multihop wireless ad hoc network testbed. *IEEE Personal Communications*, 8(1), Feb. 2001.

[19] Merit Network Inc. Multi-threaded routing toolkit. MRT Programmers Guide. http://www.merit.edu/mrt/mrt_doc/.

[20] C. Perkins and P. Bhagvat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM Computer Communications Review*, pages 234–244, Oct. 1994.

[21] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV) routing. draft-ietf-manet-aodv-06.txt, July 2000.

[22] R. Ramanathan and R. Hain. An ad hoc wireless testbed for scalable, adaptive QoS support. In *Proceedings of IEEE WCNC'2000*, Chicago, IL, USA, 2000.

[23] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, 1996.

[24] E. M. Royer and C. Perkins. An implementation study of the AODV routing protocol. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, Chicago,IL, Sept. 2000.

[25] E. M. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, Apr. 1999.

[26] C.-K. Toh. Associativity-based routing for ad-hoc mobile networks. *Wirelss Personal Communications Journal*, 4(2), Dec. 1997.

[27] C.-K. Toh and M. Delawar. Implementation and evaluation of an adaptive routing protocol for infrastructureless mobile networks. In *IEEE International Conference on Computer Communications and Networks (ICCCN'2000)*, Las Vegas, USA, Oct. 2000.

[28] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, Sept. 1991.

[29] S.-C. M. Woo and S. Singh. Longest life routing protocol (LLRP) for ad hoc networks with highly mobile nodes. In *Proceedings of IEEE WCNC'2000*, Chicago, IL, USA, 2000.